

**PROBLEMA: Quoziente di due numeri reali qualsiasi
(SOTTOPROGRAMMI)**

Scrivere un algoritmo che sia in grado di mostrare a video il valore del **quoziente** di **due** numeri **reali** qualsiasi **x** e **y** calcolato prima attraverso la procedura **QuozienteP()**, poi attraverso la funzione **QuozienteF()**.

Esempio:

Se all'interno della procedura **main()** l'utente dovesse inserire i seguenti valori:

$$\mathbf{x = 6 \quad e \quad y = 4}$$

allora:

- dopo la chiamata al sottoprogramma di tipo **PROCEDURA QuozienteP()** dovrò ottenere **1.5**;
- dopo la chiamata al sottoprogramma di tipo **FUNZIONE QuozienteF()** dovrò ottenere ugualmente **1.5**.

N.B. In una divisione occorre stare attenti principalmente al valore del divisore (ossia la variabile **y**) che, per ovvie ragioni, non potrà mai assumere il valore 0

TABELLE DEI DATI

DATI DI INPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
x	REAL	STATICA	TUTTI	Dividendo acquisito da tastiera
y	REAL	STATICA	y ≠ 0	Divisore acquisito da tastiera

DATI DI OUTPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
quoz	REAL	STATICA	TUTTI	Risultato del quoziente di x e di y mostrato a video, calcolato prima con la procedura DivisioneP() e poi con la funzione DivisioneF()

DATI DI ELABORAZIONE o DI LAVORO DEL PROBLEMA PRINCIPALE PROCEDURA main()				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI INPUT DEL SOTTOPROBLEMA: PROCEDURA QuozienteP()

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
a	INT	STATICA	TUTTI	Primo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVIDENDO
b	INT	STATICA	b ≠ 0	Secondo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVISORE supposto NON NULLO

DATI DI OUTPUT DEL SOTTOPROBLEMA: PROCEDURA QuozienteP()

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
q	INT	STATICA	TUTTI	Terzo parametro FORMALE, PASSATO PER RIFERIMENTO, che conterrà il valore del quoziente della divisione tra DIVIDENDO e DIVISORE

DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: PROCEDURA QuozienteP()

Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI INPUT DEL SOTTOPROBLEMA: FUNZIONE QuozienteF()

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
a	INT	STATICA	TUTTI	Primo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVIDENDO
b	INT	STATICA	b ≠ 0	Secondo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVISORE supposto NON NULLO

DATI DI OUTPUT DEL SOTTOPROBLEMA: FUNZIONE QuozienteF()

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: FUNZIONE QuozienteF()

Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
q	INT	STATICA	TUTTI	Variabile che conterrà il valore del quoziente della divisione tra DIVIDENDO e DIVISORE da restituire nel nome della funzione

ALGORITMO Quoziente_2_Reali_SOTTOPROGRAMMA**PROCEDURA** main ()x, y, quoz : **REAL****INIZIO**

Leggi(x)

RIPETI

Leggi(y)

FINCHE' (y ≠ 0)

/* Chiamata alla PROCEDURA DivisioneP() */

QuozienteP (x, y, quoz)

Scrivi(quoz)

/* Chiamata alla FUNZIONE DivisioneF() */

quoz ← **QuozienteF**(x, y)

Scrivi(quoz)

RITORNA**FINE****PROCEDURA** **QuozienteP** (VAL a : **REAL**, VAL b : **REAL**, REF q : **REAL**)**INIZIO**

q ← a / b

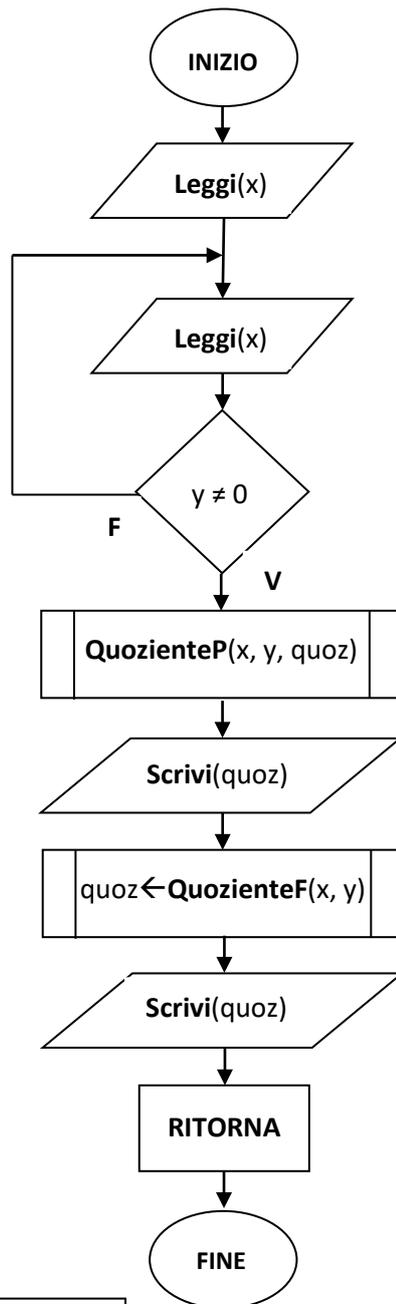
RITORNA**FINE****FUNZIONE** **QuozienteF** (VAL a : **REAL**, VAL b : **REAL**) : **REAL**q : **REAL****INIZIO**

q ← a / b

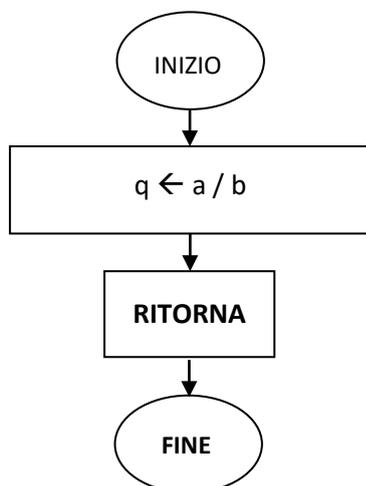
RITORNA (q)**FINE**

PROCEDURA main()

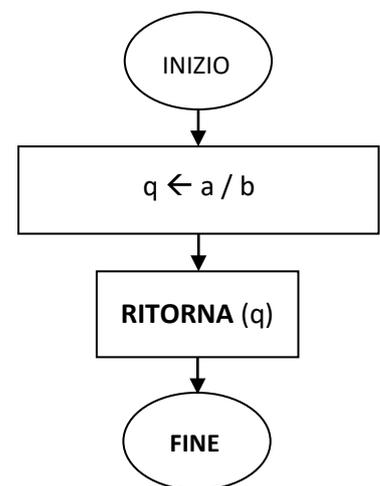
FLOWCHART



PROCEDURA QuozienteP()



FUNZIONE QuozienteF()



CODIFICA C

```
#include <stdio.h>
#include <stdlib.h>

// N.B. ESCLUSIVAMENTE nei prototipi può essere omissa il NOME DEI PARAMETRI FORMALI, NON IL TIPO
// definizione dei prototipi delle funzione QuozienteP (teoricamente è una PROCEDURA)
void QuozienteP(float a, float b, float*q);
// definizione dei prototipi delle funzione QuozienteF (teoricamente è una FUNZIONE)
float QuozienteF(float a, float b);

////////////////////////////////////
//     FUNZIONE main()           //
////////////////////////////////////
int main(int argc, char*argv[])
{
    /* dati di input */
    float x, y;

    /* dati di output */
    float quoz;

    printf("Inserisci il DIVIDENDO: ");
    scanf("%f", &x);
    do
    {
        printf("Inserisci il DIVISORE (non nullo): ");
        scanf("%f", &y);
    }
    while (y == 0);

    //CALL alla funzione QuozienteP() (teoricamente è una PROCEDURA)
    QuozienteP(x, y, &quoz);
    printf("Il quoziente calcolato con la PROCEDURA QuozienteP() e': %.2f\n", quoz);
    //CALL alla funzione QuozienteF() (teoricamente è una FUNZIONE)
    quoz = QuozienteF(x, y);
    printf("Il quoziente calcolato con la FUNZIONE QuozienteF() e': %.2f\n", quoz);

    return 0;
}

////////////////////////////////////
//     FUNZIONE QuozienteP()     // // (teoricamente è una PROCEDURA)
////////////////////////////////////
void QuozienteP(float a, float b, float*q)
{
    *q = a / b;
    return;
}

////////////////////////////////////
//     FUNZIONE QuozienteF()     // // (teoricamente è una FUNZIONE)
////////////////////////////////////
float QuozienteF(float a, float b)
{
    //dati di lavoro
    float q;

    q = a / b;
    return (q);
}
```